64000

HP64000 Logic Development System

Model 64845AF Assembler Supplement 68000/68008/68010

HEWLETT PACKARD

CERTIFICATION

Hewlett-Packard Company certifies that this product met its published specifications at the time of shipment from the factory. Hewlett-Packard further certifies that its calibration measurements are traceable to the United States National Bureau of Standards, to the extent allowed by the Bureau's calibration facility, and to the calibration facilities of other International Standards Organization members.

WARRANTY

This Hewlett-Packard system product is warranted against defects in materials and workmanship for a period of 90 days from date of installation. During the warranty period, HP will, at its option, either repair or replace products which prove to be defective.

Warranty service of this product will be performed at Buyer's facility at no charge within HP service travel areas. Outside HP service travel areas, warranty service will be performed at Buyer's facility only upon HP's prior agreement and Buyer shall pay HP's round trip travel expenses. In all other cases, products must be returned to a service facility designated by HP.

For products returned to HP for warranty service, Buyer shall prepay shipping charges to HP and HP shall pay shipping charges to return the product to Buyer. However, Buyer shall pay all shipping charges, duties, and taxes for products returned to HP from another country.

HP warrants that its software and firmware designated by HP for use with an instrument will execute its programming instructions when properly installed on that instrument. HP does not warrant that the operation of the instrument, or software, or firmware will be uninterrupted or error free.

LIMITATION OF WARRANTY

The foregoing warranty shall not apply to defects resulting from improper or inadequate maintenance by Buyer, Buyer-supplied software or interfacing, unauthorized modification or misuse, operation outside of the environment specifications for the product, or improper site preparation or maintenance.

NO OTHER WARRANTY IS EXPRESSED OR IMPLIED. HP SPECIFICALLY DISCLAIMS THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

EXCLUSIVE REMEDIES

THE REMEDIES PROVIDED HEREIN ARE BUYER'S SOLE AND EXCLUSIVE REMEDIES. HP SHALL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER BASED ON CONTRACT, TORT, OR ANY OTHER LEGAL THEORY.

ASSISTANCE

Product maintenance agreements and other customer assistance agreements are available for Hewlett-Packard products.

For any assistance, contact your nearest Hewlett-Packard Sales and Service Office.

FOLD HERE

HEWLETT PACKARD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD
FIRST CLASS PERMIT NO. 1303 COLORADO SPRINGS, COLORADO

POSTAGE WILL BE PAID BY ADDRESSEE

HEWLETT-PACKARD
Logic Product Support Dept.
Attn: Technical Publications Manager
Centennial Annex - D2
P.O. Box 617
Colorado Springs, Colorado 80901-0617

FOLD HERE

Your cooperation in completing and returning this form will be greatly appreciated. Thank you.

READER COMMENT SHEET

Operating Manual, Model 64845A
Assembler Supplement 68000/68008/68010
64845-90904, August 1984

Your comments are important to us. Please answer this questionaire and return it to us. Circle the number that best describes your answer in questions 1 through 7. Thank you.

1. The information in this book is complete:

Doesn't cover enough (what more do you need?)	1	2	3	4	5	Covers everything

2. The information in this book is accurate:

Too many errors	1	2	3	4	5	Exactly right

3. The information in this book is easy to find:

I can't find things I need	1	2	3	4	5	I can find info quickly

4. The Index and Table of Contents are useful:

Helpful	1	2	3	4	5	Missing or inadequate

5. What about the "how-to" procedures and examples:

No help	1	2	3	4	5	Very helpful
Too many now	1	2	3	4	5	I'd like more

6. What about the writing style:

Confusing	1	2	3	4	5	Clear

7. What about organization of the book:

Poor order	1	2	3	4	5	Good order

8. What about the size of the book:

too big/small	1	2	3	4	5	Right size

Comments: _____

_____

_____

_____

Particular pages with errors?

_____

Name (optional): _____
Job title: _____
Company: _____
Address: _____

Note: If mailed outside U.S.A., place card in envelope. Use address shown on other side of this card.

FOLD HERE

HEWLETT PACKARD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 1303 COLORADO SPRINGS, COLORADO

POSTAGE WILL BE PAID BY ADDRESSEE

HEWLETT-PACKARD

Logic Product Support Dept.
Attn: Technical Publications Manager
Centennial Annex - D2
P.O. Box 617
Colorado Springs, Colorado 80901-0617

FOLD HERE

Your cooperation in completing and returning this form will be greatly appreciated. Thank you.

READER COMMENT SHEET

Operating Manual, Model 64845A
Assembler Supplement 68000/68008/68010
64845-90904, August 1984

Your comments are important to us. Please answer this questionaire and return it to us. Circle the number that best describes your answer in questions 1 through 7. Thank you.

1. The information in this book is complete:

Doesn't cover enough (what more do you need?)	1	2	3	4	5	Covers everything

2. The information in this book is accurate:

Too many errors	1	2	3	4	5	Exactly right

3. The information in this book is easy to find:

I can't find things I need	1	2	3	4	5	I can find info quickly

4. The Index and Table of Contents are useful:

Helpful	1	2	3	4	5	Missing or inadequate

5. What about the "how-to" procedures and examples:

No help	1	2	3	4	5	Very helpful
Too many now	1	2	3	4	5	I'd like more

6. What about the writing style:

Confusing	1	2	3	4	5	Clear

7. What about organization of the book:

Poor order	1	2	3	4	5	Good order

8. What about the size of the book:

too big/small	1	2	3	4	5	Right size

Comments: _____

_____

_____

_____

Particular pages with errors?

_____

Name (optional): _____
Job title: _____
Company: _____
Address: _____

Note: If mailed outside U.S.A., place card in envelope. Use address shown on other side of this card.

HEWLETT PACKARD

OPERATOR/REFERENCE MANUAL

ASSEMBLER SUPPLEMENT

68000/68008/68010

© COPYRIGHT HEWLETT-PACKARD COMPANY 1980, 1981, 1984
LOGIC SYSTEMS DIVISION
COLORADO SPRINGS, COLORADO, U.S.A.

ALL RIGHTS RESERVED

Manual Part No. 64845-90904

PRINTED: **August 1984**

PRINTING HISTORY

Each new edition of this manual incorporates all material updated since the previous edition. Manual change sheets are issued between editions, allowing you to correct or insert information in the current edition.

The part number on the back cover changes only when each new edition is published. Minor corrections or additions may be made as the manual is reprinted between editions.

First Printing ..... July 1980
Second Edition ..... July 1981 (Part Number 64845-90903)
Revised ..... February 1984 (Part Number 64845-90904)
Third edition ..... August 1984 (Part Number 64845-90904)

TABLE OF CONTENTS

Chapter 1: GENERAL INFORMATION

INTRODUCTION 1-1
PROGRAMMING CONSIDERATIONS 1-1
- Microprocessor Architecture 1-1
- Address and Data Registers 1-1
- Program Counter 1-2
- Status Register 1-2
- Condition Code Register 1-2
- Source and Destination Function Code Registers 1-2
- Vector Base Register 1-2

EFFECTIVE ADDRESSING MODES 1-3
- General 1-3
- Register Direct Addressing 1-4
- Memory Addressing 1-4
- Special Addressing 1-6

CONDITION FLAGS 1-7
- Carry (C) Flag 1-7
- Overflow (V) Flag 1-7
- Zero (Z) Flag 1-7
- Negative (N) Flag 1-8
- Extend (X) Flag 1-8
- Interrupt Mask Register 1-8
- Supervisor (S) Flag 1-8
- Trace Mode (T) Flag 1-8

Chapter 2: RULES AND CONVENTIONS

OPERATIONAL INFORMATION 2-1
OPERAND INFORMATION 2-2
- Operand Location 2-2
- Dual Operands 2-2
- Register Information 2-2

ADDRESSING MODE CONVENTIONS 2-3
- Memory Address 2-3
- Special Address Modes 2-4

Chapter 3: SPECIAL PSEUDO INSTRUCTIONS

INTRODUCTION 3-1
- DC Pseudo 3-2
- DS Pseudo 3-3

TABLE OF CONTENTS (Cont'd)

Chapter 4: INSTRUCTION SET SUMMARY 68000/68008/68010

LIST OF TABLES

4-1. Abbreviation Conventions 4-1
4-2. Addressing Mode Descriptions 4-2
4-3. Instruction Set Summary 4-3

Chapter 1

GENERAL INFORMATION

INTRODUCTION

This chapter contains general information about the 68000, 68008, and 68010 microprocessors. It briefly discusses the microprocessors' architecture, addressing modes, and condition codes. For detailed descriptions of the microprocessors and their instruction set, refer to the manufacturer's User's Manual.

NOTE

If you are unfamiliar with assembly language or assemblers, read Chapter 6 in the Assembler/Linker Reference Manual. That chapter reviews, briefly, assemblers, assembly language, and the numbering systems.

PROGRAMMING CONSIDERATIONS

Microprocessor Architecture

There are 21 registers in the microprocessor for control of external memory and peripheral devices that may be associated with the target system. These registers are discussed briefly in the following paragraphs. The 68000 and 68010 have a 16-bit wide data bus, while the 68008 has an 8-bit wide data bus. The 68010 has some added features to facilitate virtual memory as noted below.

NOTE

Use the applicable processor number for the assembler directive; i.e., "68000", "68008", or "68010".

Address and Data Registers

The microprocessor has sixteen 32-bit registers that may be used for addressing or data manipulation. Eight registers (D0-D7) are assigned as data registers. The second group of registers (A0-A6) and the system stack pointer (A7) are used as address registers and stack pointers.

NOTE

Register A7 is a dual purpose stack pointer. If is either the supervisor stack pointer (SP) or the user stack pointer (USP), depending on the state of the Supervisor flag in the status register.

Any or all of the sixteen registers may be used as Index Registers.

Program Counter

The program counter register (24 bits) provides a memory address range of more than 16 megabytes. Twenty-three bits are applied directly to the address bus.

Status Register

The status register (16 bits) contains five condition flags that may be tested and used for conditional branching. In addition, the status register contains the interrupt mask register (3 bits), the trace mode flag, and the supervisor flag. These flags and their functions are discussed later in this chapter. In the 68010, the status register is only available in the supervisory mode.

Condition Code Register

The condition code register (16 bits) is the condition code half of the status register available in the user mode only in the 68010.

Source and Destination Function Code Registers

The source and destination function code registers (16 bits) allow control over the special addressing modes of the 68010.

Vector Base Register

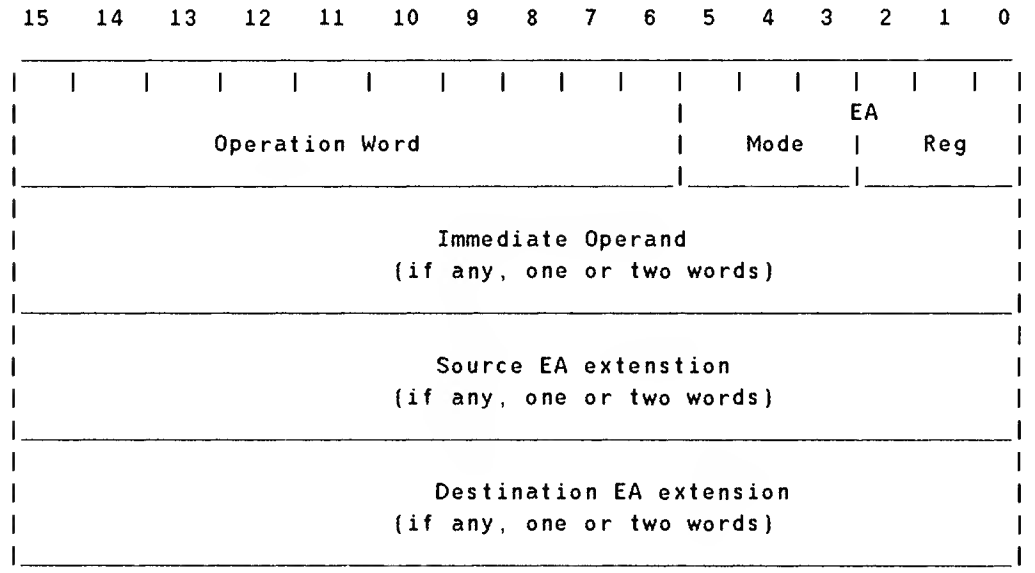
The vector base register (32 bits) points to the base location of the vector exception table in the 68010. In the 68000 and 68008 this register does not exist and the table defaults to 0H.

EFFECTIVE ADDRESSING MODES

General

Instructions may require from one to five words. Normally, the first word contains an effective address field that indicates the location of an operand. This field (EA) is composed of two 3-bit subfields: the 'mode' subfield which assigns the required address mode, and the 'register' subfield which indicates the selected register.

The EA field may require additional information to completely specify the operand. This effective address extension will be contained in a following word (or words) that will be considered as part of the instruction. The format for object code generation is as follows:



The mode subfield of the effective address may be used to indicate register direct addressing, memory addressing, or special addressing. These addressing modes are discussed in the following paragraphs.

Register Direct Addressing

In this mode of addressing, the operation will be performed directly on the content of the designated register. The types of operation are:

```
NAME                          OPERATION

Data Register Direct          EA = Dn

Address Register Direct       EA = An

Status Register Direct        EA = SR or CCR

Control Register Direct       EA = SFC,DFC,USP,VBR
```

```
  where:  An    = Address Register
         *CCR   = Condition Code Register
         *DFC   = Destination Function Code Register
          Dn    = Data Register
          EA    = Effective Address
         *SFC   = Source Function Code Register
          SR    = Status Register
         *USP   = User Stack Pointer
         *VBR   = Vector Base Register

  * only available on 68010
```

Memory Addressing

Register Indirect. In this mode of addressing, the address of the data required for the operation will be contained in an address register listed in the operand field. An indirect register address will be indicated by an address register number (A0 through A7) that is enclosed in brackets. The types of operation are:

```
NAME                                OPERATION

Register Indirect                   EA = [An]

Post-increment Register Indirect    EA = [An]; An <-- [An]+Q

Pre-decrement Register Indirect     An <-- [An]-Q; EA = [An]

Register Indirect with Offset       EA = [An]+D16

Indexed Register Indirect           EA = [An]+[Xn]+D8
   with Offset
```

```
where:    An = Address Register
          D8 = Eight-bit displacement (offset)
         D16 = Sixteen-bit displacement (offset)
          EA = Effective Address
           A = 1 for byte; 2 for word; and 4 for
                    doubleword
          Xn = Address or Data Register used
                    as an Index Register
         [ ] = Content of
         --> = Replaces
```

Post-incrementing Register Indirect. In this mode of addressing, the content of an address register will be incremented after the address is used. The increment requirement will be specified by placing a plus (+) sign immediately after the closing bracket that indicates the register indirect mode of addressing. The content of the address register will be increased by 1, 2, or 4 depending on whether the operand was byte, word, or doubleword. If the address register was the stack pointer, the address will be increased by 2 for byte and word, and 4 for doubleword. This ensures that the stack pointer remains on a word boundary.

Pre-decrementing Register Indirect. In this mode of addressing, the content of an address register will be decreased before the address is used. This decrement requirement will be specified by placing a minus (-) sign immediately before the bracket that indicates the register indirect mode of addressing. The content of the address register will be decreased by 1, 2, or 4, depending on whether the operand was byte, word, or doubleword. If the address register was the stack pointer, the address will be decreased by 2.

Offset Register Indirect. In this mode of addressing, a signed 16-bit integer offset (displacement) will be added to the content of the address register. This address mode requires one word of effective address extension.

Index Register Indirect. In this mode of addressing, a data or address register may be used as an index register and its content will be added to the content of a designated address register. This address mode requires one word of effective address extension.

Special Addressing

Absolute Data. In this mode of addressing, the memory location of the operand will be contained in the instruction itself. Operands for this mode of addresing will be indicated by a numerical quantity. The types of operation are:

```
NAME                          OPERATION

Absolute Short                EA = [PC+2]

Absolute Long                 EA = [PC+4]

where       EA = Effective Address
            PC = Program Counter
           [ ] = Content of
```

Immediate Data. In this mode of addressing, the operand of the instruction contains the value that will be used in the operation or computation. To select this mode of addressing, the corresponding operand must be preceded by the pound (#) symbol. The types of operation are:

```
NAME                          OPERATION

Immediate                     DATA = Next Word(s)

Quick Immediate               Inherent Data
```

Program Counter Relative. In this mode of addressing, the address of a memory location will be selected by adding the content of the program counter register to an offset value contained in the operand. The types of operation are:

```
NAME                          OPERATION

Relative with Offset          EA = [PC]+D16

Relative with Index           EA = [PC]+[Xn]+D8
     and Offset

Implied PC Relative           EA = [PC]+D8 or D16

where:   D8  = Eight-bit displacement (offset)
         D16 = Sixteen-bit displacement (offset)
         EA  = Effective Address
         PC  = Program Counter
         Xn  = Address or Data Register used
               as an Index Register
         [ ] = Content of
```

Program Counter with Offset. In the program counter relative addressing mode, an offset (displacement) will be added to the content of the program counter to produce an address.

Program Counter with Index. In the program counter relative addressing mode, a data or address register will be used as an index register and its content will be added to the content of the program counter to produce an address.

CONDITION FLAGS

There are five condition flags, an interrupt mask register, and two status indicators associated with the status register. The condition of each flag will be determined after the execution of certain instructions. Unless the description of a specific instruction states otherwise, the condition flags are affected as described in the following paragraphs.

Carry (C) Flag

If the result of the instruction was a carry (from addition) or a borrow (from subtraction) out of the high-order bit, the carry/ borrow flag will be set to 'one'; otherwise, it will be reset. The carry flag is assigned to bit-position 0 in the status register.

Overflow (V) Flag

For signed two's complement arithmetic operation, the overflow flag will be set if the two's complement number in the associated register exceeds the maximum number that can be represented by the two's complement notation. The overflow flag is assigned to bit-position 1 in the status register.

Zero (Z) Flag

If the result of an instruction has the value of 'zero', the zero flag will be set to 'one'; otherwise, it will be reset. The zero flag is assigned to bit-position 2 in the status register.

Negative (N) Flag

Negative numbers are expressed in two's complement form with the most significant bit (MSB) indicating the negative quality. Immediately after an operation, the processor will look at the MSB to determine if the result was negative. If so, the condition flag (N) will be set (N=1). The condition flag will be reset (N=0) if the MSB was zero. The negative flag is assigned to bit-position 3 in the status register.

Extend (X) Flag

The extend flag will be used as an operand for precision computations. When affected, it will be set the same as the carry (C) flag. The extend flag is assigned to bit-position 4 in the status register.

Interrupt Mask Register

The interrupt mask register (3 bits) permits control of up to eight levels for interrupts. The interrupt mask register is assigned bit positions 8 through 10 in the status register.

Supervisor (S) Flag

This register indicates the current mode of operation (Supervisor/ User). The SUPERVISOR mode permits certain protected operations within the microprocessor system. The SUPERVISOR flag is assigned to bit-position 13 in the status register.

Trace Mode (T) Flag

The trace mode of operation permits instruction-by-instruction tracing for program debugging. The trace mode will be available when the microprocessor is in the SUPERVISOR state as well as the USER state but may only be entered while in the SUPERVISORY mode. The trace mode condition flag is assigned to bit-position 15 in the status register.

Chapter 2

RULES AND CONVENTIONS

OPERATIONAL INFORMATION

While some instructions permit only one data field size (byte, word, or doubleword), other instructions permit variable sizes. The size code need not be specified for those instructions that have only one data field size. For those instructions that can handle more than one field size, a specific data-size code must be assigned; otherwise, the assembler defaults to size 'word'. The data size code will be designated by a period (.) immediately following the operation code and the letter B, W, or L appended, where:

```
B = byte (8-bit data)
W = word (16-bit data)
L = doubleword (32-bit data)
```

Examples:

```
ADD.B           SAM,D1

ADD             D2,D3           ;DEFAULT IS LOW ORDER
                                ;WORD OF D2,D3

        or

ADD.W           D2,D3

ADD.L           SAM,D1
```

NOTE

The data-size code may not be used with those instructions that have no size field in their instruction format.

OPERAND INFORMATION

The type of information that may be placed in the operand field will depend on the instruction, addressing mode, and register selection. Some general information concerning operands is given in the following paragraphs.

Operand Location

Instructions specify operand location by:

a. Effective Address - use of the different EA modes.

b. Implied Address - the meaning of certain instructions implies the use of specific registers.

c. Register Address - the designated register will be specified in the register field of the instruction.

Dual Operands

When two operands are used within a source statement, they must be separated by a comma. Generally, the first operand will specify the source and the second operand will specify the destination.

Register Information

Data Registers. Data registers are arranged so that byte operands will fill the low-order 8 bits, word operands will fill the low-order 16 bits, and doubleword operands will fill the entire 32 bits. When a data register is used as a source or destination operand, only the required low-order word will be altered; the remaining high-order word will be ignored.

Address Registers. The stack pointer and the seven address registers use a full 32-bit address. These registers will not support byte-size operands. An address register that specifies a source operand, will contain a word or doubleword. When used to specify a destination operand, the entire register (32 bits) will be affected.

ADDRESSING MODE CONVENTIONS

Memory Address

Address Register Indirect. The operand address will be in the address register specified by the register field in the instruction format. An indirect address will be indicated in the instruction by an address register number (A0 through A7) that is enclosed in brackets.

Example:

```
SUB             [A5],D4
```

Address Register Indirect with Post-increment. The operand address will be in the address register specified by the register firle in the instruction format. After use, the operand address will be incremented by one, two, or four, depending upon the size of the operand. The incremental requirement will be specified by placing a plus (+) symbol immediately after the closing bracket that indicates the register indirect mode of addressing.

Example:

```
MOVE            [A5]+,D4
```

Address Register Indirect with Pre-decrement. The operand address will be in the address register specified by the register field in the instruction format. Before use, the operand address will be decremented by one, two, or four, depending upon the size of the operand. The decremental requirement will be specified by placing a minus (-) symbol immediately before the bracket that indicates the register indirect mode of addressing.

Example:

```
MOVE            -[A5],D4
```

Address Register Indirect with Displacement. The operand address will be the sum of the address in the address register and the sign- extended displacement.

Example:

```
MOVE            #0FFH,10[A5]
```

Address Register Indirect with Index. The operand address will be the sum of the address in the address register, the sign-extended displacement, and the content of the designated index register.

Example:

```
MOVE            D3,10H[A4,A5]
```

Special Address Modes

Absolute Short Address. The 16-bit operand address will be sign extended before it is used. The address range is 0000H through 7FFFH.

Absolute Long Address. The operand address will be the 32-bit value specified.

Program Counter with Displacement. The operand address will be the sum of the program counter address and the sign-extended displacement. The assembler calculates this displacement by subtracting the program counter value from the operand field value.

Example:

```
BEQ             $+SAM
```

Program Counter with Index. The address will be the sum of the program counter, the sign-extended displacement, and the content of the designated index register.

Example:

```
MOVE            SAM[D4.L],CHARLIE
```

Immediate Data. An absolute number may be specified as an operand by immediately preceding a decimal or hexadecimal number with a pound (#) symbol. The symbol (#) will be used to designate an absolute number other than a displacement or an absolute address.

Example:

```
MOVE            #0FH,D4
```

Special Addressing Mode Syntax

The program counter relative addressing mode can be forced on any instruction where it is legal as follows:

Example:

```
INSTR  LABEL[PC]
INSTR  LABEL[PC,Rn]
```

This is equivalent to the following:

```
RORG
INSTR     LABEL
INSTR     LABEL[Rn]
NO_RORG
```

VARIATIONS OF INSTRUCTION TYPES

The following instructions will default to the following opcodes depending on the operand field structure.

ADD	AND	CMP	EOR	MOVE	OR	and SUB
to	to	to	to	to	to	to
ADDA	ANDI	CMPA	EORI	MOVEA	ORI	SUBA
or		or				or
ADDI		CMPI				SUBI

The ADD, MOVE, and SUB instructions will not default to ADDQ, MOVEQ, or SUBQ regardless of the operand field structure.

Chapter 3

SPECIAL PSEUDO INSTRUCTIONS

INTRODUCTION

This chapter supplements Chapter 3 in the HP Model 64000 Assembler /Linker Reference Manual. It lists and defines in detail those assembler instructions that are applicable to the 68000, 68008, and 68010 microprocessors.

DC

Define Constant

SYNTAX:

```
LABEL           OPERATION       OPERAND         COMMENT

[Name]            DC.B          expression      ;Define byte

                   or

[Name]            DC.W          expression      ;Define word

                   or

[Name]            DC.L          expression      ;Define
                                                 ;doubleword
```

The DC instruction will store a constant in memory starting with the current setting of the program counter. The instruction may contain more than one operand with each operand separated from the others by a comma. The operands may be symbols, expressions, or numbers that the assembler can evaluate to numerical values. The constant will be aligned on a word boundary if word (W) or doubleword (L) is specified, or on a byte boundary if byte (B) is specified.

The label name is optional. If present, it will be assigned the starting value of the program counter, and will reference the first constant stored by the instruction.

DS

Define Storage

SYNTAX:

```
LABEL          OPERATION      OPERAND        COMMENT

[Name]           DS.B         expression     ;Define storage
                                             ;in bytes

                 or

[Name]           DS.W         expression     ;Define storage
                                             ;in words

                 or

[Name]           DS.L         expression     ;Define storage
                                             ;in doublewords
```

The DS instruction may be used to define a block of memory. The value of the expression in the operand field specifies the number of bytes, words, or doublewords to be reserved.

Any symbol appearing in the operand field must be predefined. If the value of the operand expression is zero, no memory will be reserved; however, if the optional label name is present, it will be assigned the current value of the program counter.

ABSOLUTE_SHORT
ABSOLUTE_LONG

Absolute Addressing

SYNTAX:

```
LABEL          OPERATION          OPERAND          COMMENT

[Name]         ABSOLUTE_SHORT
                  ABS_SHORT
                     or
[Name]         ABSOLUTE_LONG
                  ABS_LONG
```

The 68000 assembler will always default to the long absolute addressing mode either when it encounters a forward reference to a program label or when the label is external. These pseudo instructions allow the user to force short or long absolute addressing modes and they can be used anywhere in the program module. EXCEPTION: when the label has been predefined or the address is a numeric value (FF8000H to 7FFFH) the assembler will optimize to the short absolute addressing mode.

Example:

```
            OBJECT      LINE
MEM.LOC.     CODE        NO.   LABEL  OPERATION         OPERAND  COMMENT


000000      33C4          3           MOVE              D4,DEST
000002      00000290

                          4           ABSOLUTE_SHORT
000006      31C4 0290     5            MOVE             D4,DEST

                          6           ABSOLUTE_LONG
00000A      33C4          7            MOVE             D4,DEST
00000C      00000290

                          .
                          .
000290                    16    DEST   DS.L             1
```

RORG

Program Counter Relative Addressing

SYNTAX:

```
LABEL          OPERATION          OPERAND          COMMENT

[Name]          RORG
                PC_INDEP

[Name]          NO_RORG
                PC_DEP
```

This pseudo instruction allows the user to declare a position independent program section. No arguments are allowed. The current program counter will not be affected. This pseudo is recommended in relocatable program modules. It should not be used in conjunction with absolute program modules; i.e., program modules containing ORG pseudo instructions. Program counter relative addressing modes will only be generated after an RORG pseudo instruction is encountered. Two examples are shown below. The first does not use RORG; the second is the same code with RORG. NO_RORG turns off the program counter relative addressing mode so no PC relative object code is generated.

Examples:

```
            OBJECT      LINE
MEM.LOC.     CODE        NO.   LABEL  OPERATION      OPERAND   COMMENT

00005E       4EB9        33           JSR            LABEL     ;Absolute
                                                               ;long
                                                               ;address
                                                               ;mode.
000060       00000084
                         .
                         .
000084       4E71        51    LABEL  NOP

                         33           RORG
000068       4EBA 001A   34           JSR            LABEL     ;PC
                                                               ;relative
                                                               ;address
                                                               ;mode.
                         .
                         .
000084       4E71        51    LABEL  NOP
                                      NO_RORG                  ;Turn off
                                                               ;PC
                                                               ;relative
                                                               ;address
                                                               ;mode.
```

EVEN

Set Program Counter to Even Address

SYNTAX:

```
LABEL          OPERATION

[Name]             DB
```

The EVEN pseudo instruction will increment the current program counter by one if it is odd. If it is even the pseudo is ignored.

If a label name is present, it is assigned the starting address of the program counter.

Example:

```
PROGRAM
COUNTER      LABEL      OPERATION      OPERAND

00000100                   DS.B        0,3FH,VAL
00000103      L1           EVEN
00000104                   DS.B        40H,0
```

A5_REL_ON
A5_REL_OFF

A5 Relative Addressing

SYNTAX:

```
          LABEL          OPERATION          OPERAND          COMMENT

         [name]          A5_REL_ON

         [name]          A5_REL_OFF
```

The 68000 assembler will always default to A5_REL_OFF addressing. The only time that A5_REL_ON will have an effect is when A5 indirect indexed addressing. The effect of this option is how the linker generates code. By setting A5_REL_ON, the linker will take the A5 value (input along with PROG,DATA,COMN,A5 value) and subtract it from the displacement of the address. This allows the user to specify to the linker where the data is to be loaded, and what A5 will be, so that the linker will address it properly. This is for compatibility with the 68000 Compiler COMMON mode.

Example:

```
OBJECT
 CODE      LABEL          OPERATION          OPERAND          COMMENT

          * File TEST
                         PROG
3A78 3010                MOVE               03010H,A5
          * User is responsible for setting A5
                         A5_REL_ON
322D 0000                MOVE               VAR1[A5],D1
          * This will use the special linker mode (VAR1-A5 value)
3B78 0001 0064           MOVE               1,100[A5]
          * This will use the special linker mode (100-A5 value)
3235 0000                MOVE               VAR1[A5,D0],D1
          * Note nothing special will happen to the above opcode
          * because it is A5 indirect indexed
                         DATA
          VAR1
```

A5_REL_ON
A5_REL_OFF
(Cont'd)

Example:

```
link
Object File? TEST
Library Files??
PROG,DATA,COMN,A5 value = 0000000H,0000010H,0000000H,0003010H
More Files? no
. . .

File = TEST:A68K:absolute
Record # 2 size = 9
  12 bytes starting at 0000H
3A78    4000    322D    D000    3B78    0001    D054    3235    0010
         |               |               |       |               |
    immediate data       |               |       |               |
                  (VAR1-A5 value)        |       |               |
                                immediate data |                 |
                                         (100-A5 value)          |
                                                               VAR1
```

Chapter 4

INSTRUCTION SET SUMMARY 68000/68008/68010

This chapter provides a quick reference to the 68000 family instruction mnemonics and their associated operands. Table 4-1 summarizes the Abbreviation Conventions which will be used in the chapter. Table 4-2 lists the Addressing Modes. All mnemonic instructions are summarized in Table 4-3. The instruction set is arranged in order of operation type. For detailed information refer to the manufacturer's user's guide.

Table 4-1. Abbreviation Conventions

SYMBOL	MEANING	DESCRIPTION
Dn	D0...D7	Data Register
An	A0...A7	Address Register
Rn	D0...A7	Data or Address Register
PC		Program Counter
SR		Status Register
*CCR		Condition Code half of Status Register
SP		Stack Pointer
*Cr	i.e.	Control Registers
	SFC	Source Function Code Register
	DFC	Destination Function Code Register
	USP	User Stack Pointer
	VBR	Base Register for Exception Vector Table
data		Immediate Value
displacement		Immediate Value
vector		Immediate Value
label		User Defined Label
s		Source Operand
d		Destination Operand

* only available on 68010.

In order to designate special modes, some opcodes allow the following postfixes.

.B	Byte	Size Postfix
.W	Word	Size Postfix
.L	Long	Size Postfix
.S	Short	Distance Postfix

The size postfix is allowed on all ARITHMETIC, BIT MANIPULATION, SHIFT-ROTATE, or DATA MOVEMENT OPERATIONS. The distance postfix is allowed on Bcc, CHK, TRAP, JMP, or the pseudo instructions DC or DS.

Table 4-2. Addressing Mode Descriptions

TITLE	ADDRESSING MODES D A [[- d d A A d d l n n A A [[[b b [[m n n A A A s s P P m]] n n n . . C C +]] , W L] , X X i i]]
ADDRESSING MODES	
data register direct	X . . . . .
indirect with predecrement	. . . . X . . . . .
all modes	X X X X X X X X X X X X
data alterable address modes	X . X X X X X X X . . .
alterable address modes	X X X X X X X X X . . .
alterable memory address modes	. . X X X X X X X . . .
data address modes	X . X X X X X X X X X X
indirect with postincrement	. . . X . . . .
register direct	X X . . . . .
control address modes	. . X . . X X X X X X .
indirect with displacement	. . . . . X . . . . .
control alterable address modes	. . X . X X X X X . . .
address register direct	. X . . . . .
immediate date	 X

Table 4-3. Instruction Set Summary

Mnemonic	Assembler Syntax		Addressing Modes
		----ARITHMETIC OPERATIONS----	
ADDITION			
Add Decimal with Extend	ABCD	Dy,Dx	s data register direct
	ABCD	-[Ay],-[Ax]	d indirect with predecrement
Add Binary	ADD	<ea>,Dn	s all modes
			d alterable memory address modes
Add Address	ADDA	<ea>,AN	all modes
Add Immediate	ADDI	#<data>,<ea>	data alterable address modes
Add Quick (1.8)	ADDQ	#<data>,<ea>	alterable address modes
Add Extended	ADDX	Dy,Dx	s data register direct
	ADDX	-[Ay],-[Ax]	d indirect with predecrement
SUBTRACTION			
Subtract Decimal with Extend	SBCD	Dy,Dx	s data register direct
	SBCD	-[Ay],-[Ax]	d indirect with predecrement
Subtract Binary	SUB	<ea>,Dn	s all modes
	SUB	Dn,<ea>	d alterable memory address modes
Subtract Address	SUBA	<ea>,An	all modes
Subtract Immediate	SUBI	#<data>,<ea>	data alterable address modes
Subtract Quick (1.8)	SUBQ	#<data>,<ea>	alterable address modes
Subtract with Extend	SUBX	Dy,Dx	s data register direct
	SUBX	-[Ay]-[Ax]	d indirect with predecrement

Table 4-3. Instruction Set Summary (cont'd)

Mnemonic	Assembler Syntax		Addressing Modes
	----ARITHMETIC OPERATIONS (cont'd)----		
NEGATE			
Negate Decimal with Extend	NBCD	<ea>	data alterable address modes
Negate	NEG	<ea>	data alterable address modes
Negate with Extend	NEGX	<ea>	data alterable address modes
MULTIPLICATION			
Signed Multiply	MULS	<ea>,Dn	data address modes
Unsigned Multiply	MULU	<ea>,Dn	data address modes
DIVISION			
Signed Divide	DIVS	<ea>,Dn	data address modes
Unsigned Divide	DIVU	<ea>,Dn	data address modes
COMPARISON			
Compare	CMP	<ea>,Dn	all modes
Compare Address	CMPA	<ea>,An	all modes
Compare Immediate	CMPI	#<data>,<ea>	data alterable address modes
Compare Memory	CMPM	[Ay]+[Ax]+	indirect with postincrement

Table 4-3. Instruction Set Summary (cont'd)

Mnemonic	Assembler Syntax		Addressing Modes
	----ARITHMETIC OPERATIONS (cont'd)----		
INCLUSIVE OR			
Inclusive OR Logical	OR	<ea>,Dn	s data address modes
	OR	Dn,<ea>	d alterable memory address modes
Inclusive OR Immediate	ORI	#<data>,<ea>	data alterable address modes
	ORI	#xxx,CCR or SR	
AND			
AND Logical	AND	<ea>,Dn	s data address modes
			d alterable memory address modes
AND Immediate	ANDI	#<data>,<ea>	data alterable address modes
	ANDI	#xxx,CCR or SR	
EXCLUSIVE OR			
Exclusive OR Logical	EOR	Dn,<ea>	data alterable address modes
Exclusive OR Immediate	EORI	#<data>,<ea>	data alterable address modes
	EORI	#xxx,CCR or SR	
Logical Complement	NOT	<ea>	data alterable address modes
SIGN EXTEND			
Sign Extend	EXT	Dn	data register direct

Table 4-3. Instruction Set Summary (cont'd)

Mnemonic	Assembler Syntax		Addressing Modes
	----ARITHMETIC OPERATIONS (cont'd)----		
TEST VALUE			
Test and Set an Operand	TAS	<ea>	data alterable address modes
Test an Operand	TST	<ea>	data alterable address modes
Clear an Operand	CLR	<ea>	data alterable address modes
SET VALUE			
Scc Set According to Condition CC carry clear CS carry set EQ equal GE greater or equal GT greater than HI high LE less or equal LS less or same LT less than MI minus NE not equal PL plus VC overflow clear VS overflow set	Scc	<ea>	data alterble address modes
	----BIT MANIPULATION OPERATIONS----		
Test a Bit and change	BCHG	Dn,<ea>	data alterable address modes
	BCHG	#<data>,<ea>	data alterable address modes
Test a Bit and clear	BCLR	Dn,<ea>	data alterable address modes
	BCLR	#<data>,<ea>	data alterable address modes

Table 4-3. Instruction Set Summary (cont'd)

Mnemonic	Assembler Syntax		Addressing Modes
	----BIT MANIPULATION OPERATIONS (cont'd)----		
Test a Bit and Set	BSET	Dn,<ea>	data alterable address modes
	BSET	#<data>,<ea>	data alterable address modes
Test a Bit	BTST	Dn,<ea>	data alterable address modes
	BTST	#<data>,<ea>	data alterable address modes
	----SHIFT - ROTATE OPERATIONS----		
LSL,LSR Logical Shift and Rotate			
Immediate	LSL	#<data>,Dy	data register direct
Register	LSL	Dx,Dy	data register direct
Memory	LSL	<ea>	alterable memory address modes
ROL,ROR Rotate without Extend			
Immediate	ROL	#<data>,Dy	data register direct
Register	ROL	Dx,Dy	data register direct
Memory	ROL	<ea>	alterable memory address modes
ROXL,ROXR Rotate with Extend			
Immediate	RORX	#<data>,Dy	data register direct
Register	RORX	Dx,Dy	data register direct
Memory	RORX	<ea>	alterable memory address modes
ASL,ASR Arithmetic Shift and Rotate			
Immediate	ASL	#<data>,Dy	data register direct
Register	ASL	Dx,Dy	data register direct
Memory	ASL	<ea>	alterable memory address modes

Table 4-3. Instruction Set Summary (cont'd)

Mnemonic	Assembler Syntax		Addressing Modes	
		----DATA MOVEMENT OPERATIONS---		
Exchange Registers	EXG	Rx,Ry		register direct
Load Effective Address	LEA	<ea>,An		control address modes
Link and Allocate	LINK	An,#<displacement>		address register direct
Move Data from Source to	MOVE	<ea>,<ea>	s d	all modes data alterable address modes
Destination	MOVE	<ea>,CCR or SR	s	data address modes
	MOVE	SR,<ea>	d	data alterable address
*	MOVE	CCR,<ea>		modes
	MOVE	USP,An		
	MOVE	An,USP		
Move Address	MOVEA	<ea>,An		all modes
*Move to/from	MOVEC	Rn,Cr		register direct
* Control Register	MOVEC	Cr,Rn		register direct
Move Multiple Registers	MOVEM	<reg list>,<ea>	s	control alterable address modes
		<ea>,<reg list>	d	control addrress modes
Move Peripheral Data	MOVEP	Dx,d[Ay]		indirect with displacement
	MOVEP	d[Ay],Dx		indirect with displacement
Move Quick (1..8)	MOVEQ	#<data>,Dn		data register direct
*Move to/from	MOVES	<ea>,Rn	s	alterable address modes
* Address Space	MOVES	Rn,<ea>	d	alterable address modes

* only available on 68010.

Table 4-3. Instruction Set Summary (cont'd)

Mnemonic	Assembler Syntax		Addressing Modes
	----DATA MOVEMENT OPERATIONS (cont'd).----		
Push Effective Address	PEA	<ea>	control address modes
Swap Register Halves	SWAP	Dn	data register direct
Unlink	UNLK	An	address register direct
	----PROGRAM CONTROL OPERATIONS----		
Bcc Branch Conditionally CC carry clear CS carry set EQ equal GE greater or equal GT greater than HI high LE less or equal LS less or same LT less than MI minus NE not equal PL plus VC overflow clear VS overflow set	Bcc	<label>	
Branch Always	BRA	<label>	
Branch to Subroutine	BSR	<label>	

Table 4-3. Instruction Set Summary (cont'd)

Mnemonic	Assembler Syntax		Addressing Modes
	----PROGRAM CONTROL OPERATIONS (cont'd)----		
Test Condition, Decrement, and Branch CC carry clear CS carry set EQ equal F false GE greater or equal GT greater than HI high LE less or equal LS low or same LT less than MI minus NE not equal PL plus T true VC overflow clear VS overflow set	DBcc	<label>	
Illegal Instruction	ILLEGAL		
Jump	JMP	<ea>	control address modes
Jump to Subroutine	JSR	<ea>	control address modes
No Operation	NOP		
*Return and Deallocate Parameters	RTD	#<displacement>	immediate data
Return from Subroutine	RTS		

* only available on 68000.

Table 4-3. Instruction Set Summary (cont'd)

Mnemonic	Assembler Syntax		Addressing Modes
	----SYSTEM CONTROL OPERATIONS----		
Check Register Against Bounds	CHK	<ea>,Dn	data address modes
Reset External Devices	RESET		
*Return from *Exception	RET		
Load Status Register and Stop	STOP	#xxx	immediate data
Trap	TRAP	#<vector>	
Trap on Overflow	TRAPV		

* only available on 68010.

64845-90904, AUGUST 1984
Replaces: 64845-90904, February 1984

PRINTED IN U.S.A.